

Project #4
16F877A PICMicro programming under MPLAB
Due: January 31, 2025

Instructor: Dr. Hanna Bullata

Simple 16F877A Picmicro-based game

We would like to create an interactive memory-based game that employs 4 different color LEDs, 4 push buttons and a 16 × 2 character LCD. Players replicate LED light sequences using the push buttons. The LCD will give instructions to the player on what to do.

The system should behave as follows:

1. When the system is powered up, the first line of the LCD should display the message 'Press any button' and the second line of the LCD should display the message 'to start'.

The above message blinks 3 times with a 1 second delay between blinks. Afterwards, we move to the next step.

2. The LCD should display the message 'Level 1' on the first line and a random LED pattern using 3 LEDs will take place. In future levels, a random LED pattern is generated with an increasing generated pattern as the game progresses.
 - **Level 1:** 3 LED-sequence will light up in a random sequence with a 1-second ON period and a 1-second OFF period.
 - **Level 2:** 4 LED-sequence will light up in a random sequence with a 1-second ON period and a 1-second OFF period.
 - **Level 3:** 5 LED-sequence will light up in a random sequence with a 1-second ON period and a 1-second OFF period.
 - etc ... up to level 20 where 22 LED-sequence is used.

Note that the same LED might light up multiple times in the same sequence.

3. When the sequence generated in the previous step ends, the player needs to replicate the light sequence using the corresponding push buttons. When a push button is clicked, the corresponding LED will light up as long as the push button is clicked.
4. When the player stops clicking on the push buttons for 2 seconds, the controller needs to check if the displayed LED sequence matches the player's input on the push buttons.

If there is a match, the LCD will display the message 'Congratulations' and the player will move to the next level.

If there is no match, the LCD will display the message 'Try again'.

The player is allowed to do 3 trials max before declaring that the player lost the game or move to the next level if a match is found.

5. When a game ends, the player is instructed by a message on the LCD to hit any push button to start a new game. Similarly, if the player doesn't hit any push button for over a minute, the game resets to level 1 after displaying on the LCD that the game time is over.

Sequence Example

Below is a sequence example that might give you a better idea about the game behavior.

- The player hits any push button and the following pattern is displayed on the LEDs: LED 1 ON then OFF, LED 1 ON then OFF, LED 3 ON then OFF
- The player is requested to enter the sequence he/she just witnessed using the push buttons.
- If the sequence entered by the player matches the sequence that just occurred on the LEDs, the player has completed successfully the current level and is moved to the next level which represents a bigger challenge by adding 1 extra LED lighting to the new sequence (up to level 20 that involves 22 LED sequence).
Example: LED 1 ON then OFF, LED 3 ON then OFF, LED 3 ON then OFF, LED 1 ON then OFF.

- Of course as the game involves more LED lighting (as the level becomes more complex), the player will start doing mistakes.

What you should do

- Use proteus application to build the schematic for the whole system, including the 4 colored LEDs, the 4 push buttons and the 16×2 character LCD.
- Use the 16F877A pins of your choice for the LEDs and the pins of your choice for the push buttons.
- Use the 16F877A pins of your choice to connect the character LCD in 4-bit mode. Remember to pull up the RS pin of the LCD using a 4.7K Ω resistor.
- Remember to add a 10K Ω pull-up resistors to the push buttons (or use the internal weak pull-up resistors of PortB), add a 4MHZ oscillator (with $2 \times 15\text{pF}$ capacitors) and a 10K Ω pull-up resistor to the MCLR pin.
- Enable interrupts on the push buttons (if you wish to use interrupts).
- Build the PIC assembly code for the simple 16F877A picmicro-based game that implements the behavior described above under MPLAB IDE.
- Assemble the project and make sure you get a successful build. Use the simulator if you wish to make sure the behavior is correct.
- Send the zipped folder that contains the MPLAB code for the project before the deadline. If the deadline is reached and you are still having problems with your code, just send it as is!